# 基于自适应免疫计算的网络攻击检测研究

陈晋音 徐轩桁 苏蒙蒙
(浙江工业大学信息工程学院 杭州 310023)

**摘 要** 互联网与生俱来的开放性和交互性的特征,导致攻击者能利用网络的漏洞对网络进行破坏。网络攻击一般具有隐蔽性和高危害性,因此有效地检测网络攻击变得极为重要。为了解决大部分检测算法只能检测一类网络攻击且检测延迟高等问题,提出了一种基于自体集密度自动划分聚类方法的阴性选择算法,简称 DAPC-NSA。该算法采用基于密度的聚类算法对自体训练数据进行预处理,对其进行聚类分析,剔除噪声并生成自体检测器;然后根据自我检测器生成非我检测器,同时利用自我检测器和非我检测器来检测异常。文中最后进行了模拟入侵检测实验,结果表明,相比于其他检测算法,该算法不仅能同时检测 6 种攻击,具有较高的检测率和较低的误测率,而且检测时间短,能达到实时检测的目标。

**关键词** 网络安全,攻击检测,自适应免疫,DAPC-NSA,检测器,网络攻击模拟
**中图法分类号** TP183 **文献标识码** A

## Research on Network Attack Detection Based on Self-adaptive Immune Computing

CHEN Jin-yin XU Xuan-yan SU Meng-meng
(College of Information and Engineering, Zhejiang University of Technology, Hangzhou 310023, China)

**Abstract** The Internet is inherently open and interactive, making the attacker use the network vulnerabilities to destroy the network. Network attacks are generally conceal and highly hazardous, so how to effectively detect network attacks becomes extremely important. In order to solve the problem that most of the detection algorithms can only detect a kind of network attack, and the detection delay is high, this paper proposed a negative selection algorithm based on density automatic partition clustering method with self-set, referred to DAPC-NSA. The algorithm uses the density clustering algorithm to preprocess the self-training data, performs cluster analysis on the training data, eliminates the noise, and generates the self-detector. And then it generates the nonself-detector according to the self-detector, and uses the self-detector and nonself-detector to detect the anomalies. The simulated intrusion detection experiment was carried out. The experiment shows that the algorithm can not only detect six kinds of attacks simultaneously, but also has the higher detection rate and the lower false alarm rate. The detection time is short compared with other detection algorithm, and it can achieve the target of real-time detection.

**Keywords** Network security, Attack detection, Self-adaptive immune, DAPC-NSA, Detectors, Network attack simulation

## 1 引言

互联网具有开放性,攻击者通常利用网络漏洞来绕过网络的安全监测,并对目标网络进行一定的破坏。网络攻击的种类也多种多样,包括恶意扫描、拒绝服务(DOS)、病毒、蠕虫、数据窃取和篡改等。网络攻击会对网络的性能和安全造成很大的危害。TCP/IP 协议是当今信息网络最常用的通信协议,但是由于其在设计之初并没有过多地考虑网络安全,大部分网络攻击都是基于 TCP/IP 协议自身的缺陷而产生的,而网络攻击一般具有隐蔽性和高危害性,因此研究如何有效地检测网络攻击对保护整个互联网的安全性具有非常重要的意义。

近几十年来,研究者们对网络攻击检测进行了深入的研究[1-10]。Xiong 等[1]提出了一种基于协同神经网络和灾难理论的检测方法,其通过描述并分析网络流量的动态特征行为来检测云通信网络的异常流量 。Sperotto 等[2-3]提出了一种基于 IP 流的入侵检测方法,其分析对象是网络的数据流,而不是每个单独数据包的内容。Tan 等[4]提出了一种基于多变量相关分析(MCA)的 DOS 攻击检测系统,其通过提取网络流量特征之间的几何相关性,利用 MCA 准确表征网络流量的特征,并利用基于异常的检测原理进行攻击检测。Lee 等[8]通过研究 DDOS 攻击的过程来提取相应的特征,再进行聚类分析,区分正常流量和攻击流量,从而可以提前检测 DDOS 攻击。Siris 等[9]提出并评估了两种用于检测 TCP SYN FLOOD 攻击的异常检测算法,即自适应阈值算法和用于变化点检测的累积和算法(CUSUM)[10]。这两种算法在高强度的攻击下,都有着很高的检测率,但在低强度攻击下的效果并不理想,误检率较高;而且,这两种算法都不能很快地检测出 TCP SYN 攻击,会有 20～30s 的延迟。

综上所述,虽然人们提出了很多针对网络攻击的检测方

本文受国家自然科学青年基金(61502423),浙江省科技厅科研院专项(2016F50047)资助。
**陈晋音**(1982－),女,副教授,主要研究方向为网络安全、数据挖掘、智能计算等;**徐轩桁**(1994－),男,硕士生,主要研究方向为网络安全。

法，但它们都存在着一些问题：大多数方法都只能检测一类攻击；算法的检测时间一般比较长，如果攻击持续时间只有40 s，结果检测出攻击就需要 20 s，那么我们就不能尽早地检测出攻击并做出防御；几乎所有的算法都是在已经存在的 DARPA 或 KDD99 入侵数据集上验证可行性和检测精度，并没有在实际网络中进行检测，因此并不能保证算法在真实网络中的可行性。

针对以上问题，本文提出了一种基于自体集密度自动划分聚类方法的阴性选择算法，简称 DAPC-NSA。该算法首先采用基于密度的聚类算法对自体训练数据进行预处理，将训练数据进行聚类分析，剔除噪声并生成自体检测器；然后根据自我检测器生成非自我检测器。实验表明，DAPC-NSA 算法确实可以排除噪声对检测器的影响，减少了自我检测器的个数，降低了程序在距离计算上的时间开销，并在一定程度上降低了实验的误判率。最后还对真实的 TCP/IP 协议数据包进行了入侵检测实验，结果表明，该方法具有较高的检测率和较低的检测延迟。

## 2　相关检测器生成算法的研究

### 2.1　NSA

和强大的信息处理能力[12]。人工免疫算法包括否定选择算法（NSA）、克隆选择算法（CSA）、免疫网络和危险理论等，其中 NSA 是人工免疫理论中的一种非常重要的检测器生成算法，它具有区别自体和异常的能力。

否定选择算法[13]首先由 Forrest 于 1994 年提出，其原理是根据自我集随机生成一系列候选检测器，并通过否定选择操作将其变成成熟的检测器后用来检测异常。$Self \subseteq U$ 表示正常样本，$Nonself \subseteq U$ 表示异常样本，并且 $Self \cup Nonself = U$，$Self \cap Nonself = 0$。检测器 $d(c, r_d)$ 用于识别异常，其中 $c \in Nonself$ 表示检测器所在的位置向量，$r_d$ 表示检测器的半径，此时与检测器 $d$ 的距离小于 $r_d$ 的样本被识别为非我。$Self$，$Nonself$，$U$ 和 $d$ 之间的关系如图 1 所示。

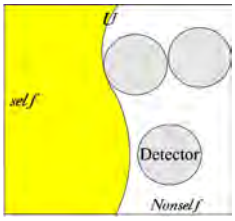



图 1　自体、非自体与检测器的分布

但是传统的实数型否定选择算法（RNSA）采用半径固定的检测器[14-15]，这会导致生成的检测器数量较多，而且会产生很多黑洞，致使检测器的覆盖率不理想。而 V-detector 算法和 Ft-NSA 算法是 NSA 改进算法的代表。

### 2.2　V-detector

为了解决固定半径的 NSA 漏洞多、检测器个数多的问题，文献[16]提出了半径可变的 V-detector 算法，它把随机样本与自我样本集的最短距离作为非我检测器的半径。该算法通过半径可变的非我检测器，在距离自我集较远的地方可以生成半径较大的非我检测器，从而减少检测器的数量；同时，在自我集边缘部分可以生成半径较小的检测器，从而提高检测率，使得检测器和自我集边缘切合得更好。

算法的具体流程如图 2 所示。V-detector 首先计算随机生成点 $t$ 与自我样本之间的距离 $dis(t,s)$，若 $dis(t,s) < r_s$（$r_s$ 为自我检测器的半径），则舍去随机生成点；否则，继续判断点 $t$ 与非我检测器的距离 $dis(t,d)$，若 $dis(t,d) < r_d$，则将其舍去，否则生成半径为 $r_t = dis(t,s) - r_s$ 的候选检测器。当随机生成点的个数达到 $n$（$n$ 为固定采样个数）时，将所有候选检测器存入非我检测器。如此不断循环，直到满足检测器的覆盖要求。图 3 是当 $p = 99\%$，$z_a = 1.28$ 时利用 V-detector 算法生成检测器的效果图。从图 3 可以看出，检测器（黄色部分）几乎覆盖了整个非我区间，检测器与自我样本集边缘切合得很好，且检测器的数量较少。

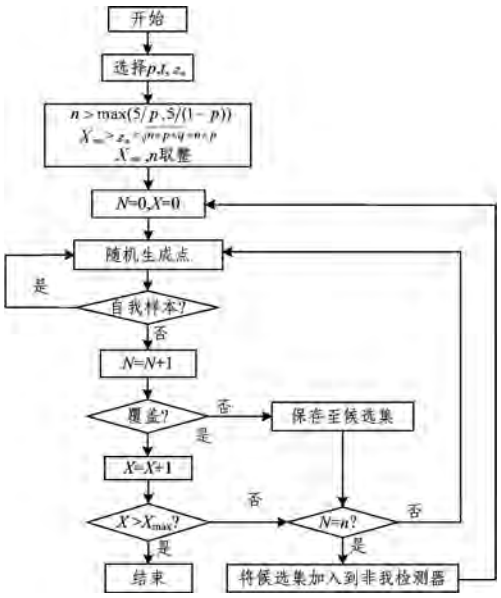
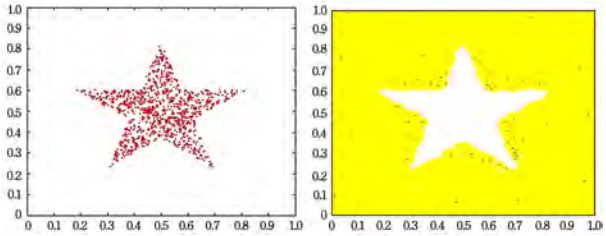



图 2　V-detector 非自体检测器生成的流程图



（a）自体样本分布空间　　（b）检测器覆盖空间

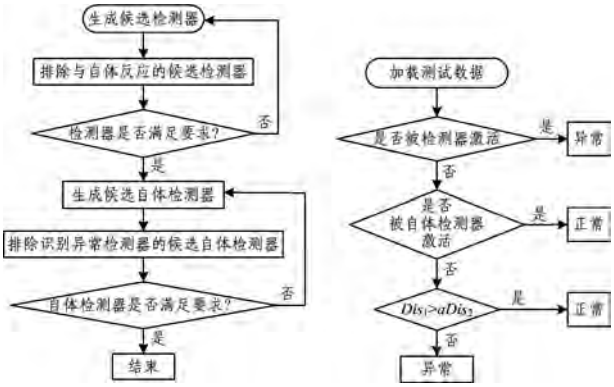
图 3　利用 V-detector 生成检测器的效果图

### 2.3　Ft-NSA

由于自我样本分布不均匀，在自我样本集内部可能会产生由于自我样本分布疏松而导致的漏洞，因此在自我样本集内产生非我检测器。为了解决此类漏洞问题并减少自我集样本的个数，Ft-NSA 在生成非我检测器之后，用类似的方法生成自我检测器。

Ft-NSA 算法的两个阶段的流程图如图 4 所示。虽然 Ft-NSA 算法解决了 V-detector 将非我检测器没有检测到的部分都归纳为正常情况而导致的高虚警率和漏洞问题，但其只是在 V-detector 的基础上增加了一个步骤，而且在训练阶段生成非我检测器和自我检测器的时间较长，在检测阶段样本检测的时间也较长，同时对自我样本的噪声也没有进行相应处理。另外，自我检测器的生成不仅会使得部分自我集被误判成非自我检测器，还会因为噪声把部分非我错判为自我。针对以上问题，本文提出了基于聚类方法先生成自我检测器再生成非我检测器的方案，从而在保证 Ft-NSA 优势的同时，

解决 Ft-NSA 的虚警率问题。




(a)训练阶段算法 (b)检测阶段算法

图 4 Ft-NSA 算法两个阶段的流程图

## 3 DAPC-NSA 检测器生成算法的设计

DAPC-NSA 算法首先根据自体集的密度峰值计算出聚类中心并删除噪声,然后根据非噪声自我样本生成自我检测器,再根据自我检测器生成非我检测器,最后同时使用自我检测器和非我检测器来检测样本是否异常。

### 3.1 DAPC 算法的主要思想

大多数的否定选择算法 NSA 都无法对噪声进行处理,噪声一旦产生,将对整个实验结果产生较大的影响。针对此类问题,本文设计了一种快速搜寻密度峰的聚类方法,其利用聚类中心的密度比周围点高并且到密度更高的点距离十分远的特点对自我样本进行处理,当某一点到聚类中心的距离超过预定值时,将其判断为噪声。为了更好地说明此方法,给出如下定义。

**定义 1** 点 $i$ 的局部密度表示 $\rho_i$ 与点 $i$ 的距离小于 $d_c$ 的点的个数。$\rho_i$ 的计算公式如式(1)所示,其中 $d_{ij}$ 表示点 $i$ 和点 $j$ 之间的欧氏距离,$d_c$ 为截断距离参数,由输入参数 $t$ 决定:

$$\rho_i=\sum_{j\neq i}\chi(d_{ij}-d_c) \tag{1}$$

但是在有些数据集中,每个点的密度估计可能会存在误差,严重时会影响算法的效果。为此,需要一种更加精确的密度计算公式:

$$\rho_i=\sum_{j\neq i}\exp(-(\frac{d_{ij}}{d_c})^2) \tag{2}$$

**定义 2** $\delta_i$ 定义为样本 $i$ 到局部密度比它大且距离最近的样本 $j$ 之间的距离,计算公式如式(3)所示。其中对于局部密度最高的样本点,$\delta_i$ 的计算公式如式(4)所示。

$$\delta_i=\min_{j:\rho_j>\rho_i}(d_{ij}) \tag{3}$$

$$\delta_i=\max_{j}(d_{ij}) \tag{4}$$

**定义 3** 聚类中心是指被密度更小的点所包围且与密度更大的点之间的距离很远的点。

**定义 4** 如果一个已分配的点与其他类中的点的距离小于截断距离 $d_c$,则该点为边界点。一个类中的所有边界点构成了边界点集。

**定义 5** $\rho_b$ 为每个类的边界中密度最高的点,类中密度高于 $\rho_b$ 的点被认为是有用数据,密度低于 $\rho_b$ 的点被认为是噪声。

在正常计算时,首先根据定义 3 得到聚类中心,即 $\delta_i$ 和 $\rho_i$ 都很大的点。在确定聚类中心以后,每个点都将被分配给与其最近的更高密度的点,从而形成多个类。但并不是所有的点都将被分配到某一个类,否则我们将无法区别噪声和正常点。我们将根据定义 5 对噪声进行处理。但是,这种噪声判断办法存在一些问题:当噪声的数量比较少,且分属每个类的噪声之间的距离都大于 $d_c$ 或只有一个类时,每个类的 $\rho_b$ 将不能被计算,即每个类的 $\rho_b$ 都为 0,亦即所有的点都是正常的,这显然是不合理的。因此,本文通过式(5)对 $\rho_b$ 再次进行计算。

$$\rho_i^b{}'=\rho_i^b+\frac{\rho_{i\max}-\rho_i^b}{hc} \tag{5}$$

其中,$\rho_i^b$ 是 $DAPC$ 算法得出的第 $i$ 个类边界的最大密度,$\rho_{i\max}$ 为第 $i$ 个类的最大密度,$\rho_i^b{}'$ 为再次计算后所获得的噪声密度阈值。在每个类中,密度高于 $\rho_i^b{}'$ 的点为有用的点,即正常的训练数据;密度低于 $\rho_i^b{}'$ 的点被认为是隐藏在训练数据中的异常样本,也就是噪声。$hc$ 为算法的输入参数,用来调节噪声密度阈值。

自体集聚类的流程图如图 5 所示。

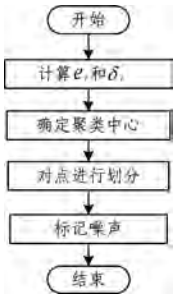


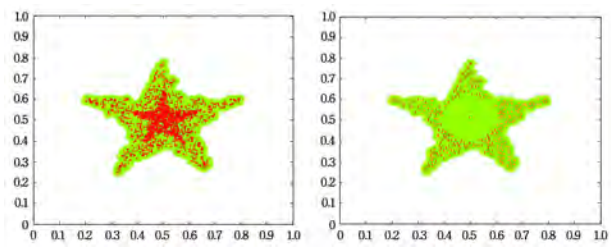

图 5 自体集聚类的流程图

### 3.2 主要检测原理

#### 3.2.1 自体检测器的生成

由于 Ft-NSA 方法和 V-detector 方法的自我样本的个数较多,在生成非我检测器时会有大量的时间消耗在距离计算上,因此本文在生成非我检测器之前先生成自我检测器,从而减少非我检测器的生成时间。首先,以每个非噪声的点为圆心,生成半径为 $R_s$($R_s$ 为 DAPC-NSA 算法的输入参数,与 V-detector 和 Ft-NSA 算法中的自体半径一致)的固定半径自体检测器。然后,以聚类中心为圆心、以聚类中心到最近的噪声点的距离为半径生成一个大圆,并且将完全镶嵌在这个大圆里的无用常半径自体检测器剔除,从而大大减少了自体检测器的个数。

实验效果如图 6 所示。



(a)固定半径 $R_s$ (b)DAPC-NSA 算法

图 6 自体检测器生成方法

图 6(a)表示采用固定半径的自体检测器,这样会导致自体检测器出现大量重叠,尤其是在训练数据密度较高的区域

内，自体检测器个数非常多，导致在生成非我检测器阶段的效率很低。图 6(b)为 DAPC-NSA 算法生成的自体检测器，以聚类中心为圆心的大圆很好地覆盖了高密度区域的自体检测器，使得自体检测器大大减少；同时，自体区域边缘的检测器个数也大大减少。图 6 表明，DAPC-NSA 算法可以很好地减少自体的个数，从而提高生成非我检测器的效率。

自体检测器生成的流程图如图 7 所示。

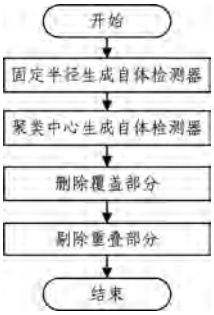



图 7　自体检测器生成的流程图

3.2.2　非我检测器的生成

无论是 Ft-NSA 还是 V-detector 算法，当 $p=99\%$ 时，算法每一次循环生成的随机点个数至少为 $n=500$。而每次循环里随机点只与先前获得的非我检测器判断是否被覆盖，而不比较 500 个随机点对应非我检测器之间的覆盖率，这使得一次循环生成的非我检测器重叠率较高，无效的检测器个数较多。而当 $p=99.9\%$ 时，随机点个数 $n=5000$，这样一来问题就更加严重了。为了解决此类问题，我们参考文献[18]提出的检测器终止条件，使得如果连续出现 $x$ 个候选检测器(未被已有的非我检测器覆盖)，则将候选检测器存入检测器，以降低非我检测器的重叠率。算法的具体流程图如图 8 所示。

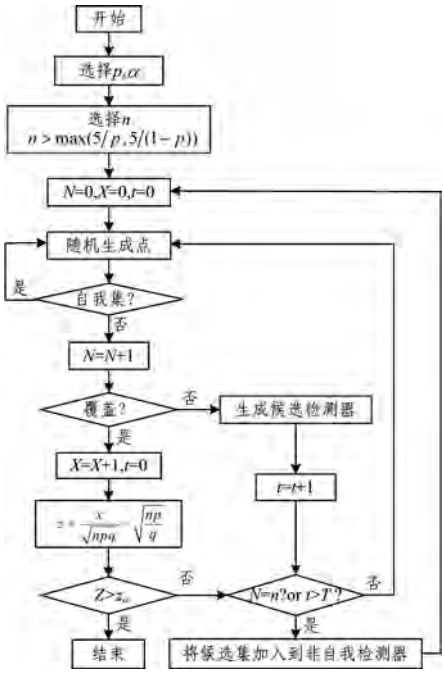



图 8　非自体检测器生成的流程图

## 4　网络攻击入侵检测实验

TCP/IP 协议是互联网最基本的协议，旨在提供简单、高效和开放的通信设施，但是它不提供认证、完整性和隐私机制，在设计之初并没有过多地考虑网络的安全性，导致攻击者可以利用漏洞发起攻击，对网络安全造成威胁。通常的入侵检测方法只能检测一类攻击，且检测延迟大，不能做到实时检测；本文提出利用否定选择算法 NSA 来检测基于 TCP/IP 协议的网络攻击，旨在实时检测多类攻击。

### 4.1　网络攻击的定义

基于 TCP/IP 协议的攻击种类很多[20]，本文主要针对以下 6 种攻击进行分析。

4.1.1　SYN FLOOD 攻击

攻击原理：攻击者利用 TCP 三次握手机制，给目标计算机发送大量 TCP SYN 报文，而没有第三次 ACK 回应；如果 TCP 半连接数很多，那么就会把目标计算机的资源耗尽，使其不能响应正常的 TCP 连接请求。

4.1.2　分片 IP 报文攻击

攻击原理：如果攻击者只给目标计算机发送一片分片报文，而不发送所有的分片报文，这样攻击者计算机便会一直等待(直到一个内部计时器到时)；如果攻击者发送了大量的分片报文，就会消耗掉目标计算机的资源，从而导致其不能响应正常的 IP 报文。这是一种 DOS 攻击。

4.1.3　泪滴攻击

攻击原理：对于一些大的 IP 包，需要进行分片传输，因此必须得对 IP 报头的偏移字段进行设置；如果攻击者把偏移字段设置成不正确的值，即可能出现重合或断开的情况，进而导致目标操作系统崩溃。

4.1.4　Land 攻击

攻击原理：用于 Land 攻击的数据包中的源地址和目标地址是相同的，因为操作系统接收到这类数据包时，不知道如何处理堆栈中通信源地址与目的地址相同的情况；或者循环发送和接收该数据包而消耗大量的系统资源，从而有可能造成系统崩溃或死机等现象。

4.1.5　WinNuke 攻击

攻击原理：WinNuke 攻击即是利用了 Windows 操作系统的一个漏洞，向这个 139 端口发送一些携带 TCP 带外(OOB)数据报文；这些攻击报文与正常携带 OOB 数据报文不同，其指针字段与数据的实际位置不符，即存在重合，这样 Windows 操作系统在处理这些数据时，就会崩溃。

4.1.6　针对 TCP 标志位的攻击

攻击原理：正常情况下，任何 TCP 报文都会设置 SYN，FIN，ACK，RST，PSH 5 个标志中的至少一个标志，第一个 TCP 报文(TCP 连接请求报文)设置 SYN 标志，后续报文都设置 ACK 标志；如果攻击者错误设置了 TCP 标志位，比如同时设置 SYN 和 FIN，目标计算机将不能处理这类畸形报文，导致系统崩溃。

### 4.2　攻击检测实验

入侵检测实验分为 3 部分：网络数据包模拟，数据包解析及特征提取，NSA 检测攻击。

4.2.1　网络数据包模拟

在实验中，为了验证算法在高强度 DOS 攻击和低强度 DOS 攻击下的性能，模拟了两种 TCP SYN FLOOD 攻击，分别是 1 秒 500 个半连接 SYN 包(高强度攻击)，以及 1 秒 40 个半连接 SYN 包(低强度攻击)，持续 11 秒。分片 IP 报文攻击也是一种 DOS 攻击，实验中只模拟了低强度的攻击，1 秒 40 个 IP 分片报文，持续 5 秒。泪滴攻击的原理是把偏移字

段设置成不正确的值，因此本文通过将生成偏移字段设置为0，1300，3000的分片IP数据包组来模拟泪滴攻击。由于Land攻击的数据包中的源地址与目标地址是相同的，因此模拟生成了源IP地址和目的IP地址相同的TCP SYN包。WinNuke攻击的特征是被攻击的目标端口通常为139，138，137，113，53，且URG标志位设置为1，本文根据该特征生成了相应的TCP数据包。此外，针对TCP标志位的攻击有很多种，比如SYN和FIN同时设置为1，没有设置任何标志，设置了FIN或PSH或URG但没有设置ACK，SYN和RST同时设置等，因此将模拟数据包中的TCP标志位设置为[0 19 3 22 6 21 1 8 32 40]中的值。正常的数据包基本是根据TCP三次握手和四次分手机制模拟出来的。

本文实验的数据包都是根据图9的格式在MATLAB上模拟出来的，模拟的具体数据都是十六进制的，如图10所示。这里，每个数据的首位特征表示数据包发送或接收的时间；中间是数据包格式；末位特征代表标志位，0表示正常数据包，1表示攻击数据包。因此，本文根据图9的数据包格式和上文提到的6种攻击模拟了某个主机在0到220 s时间内收到和发送的所有数据包(包括攻击数据包和正常数据包)。本文在模拟的数据包上进行了基于NSA算法的入侵检测实验。

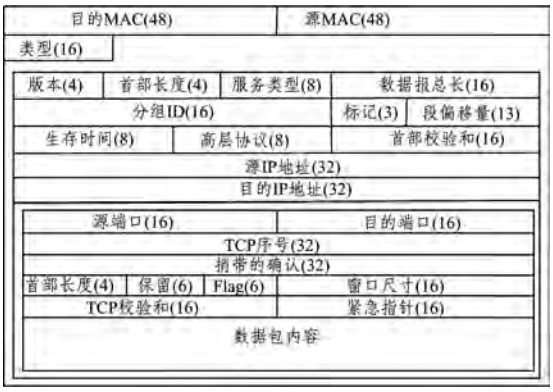



图9 TCP数据包格式

```
0.001000 14 F6 F4 6B F5 3A 20 89 84 20 8A 5C 08 00 45 00 00 28 00 01 40 00 40 06 00 00 C0
0.023044 20 89 84 20 8A 5C 14 F6 F4 6B F5 3A 08 00 45 00 00 28 00 02 40 00 40 06 00 00 76
0.023189 14 F6 F4 6B F5 3A 20 89 84 20 8A 5C 08 00 45 00 00 28 00 01 40 00 40 06 00 00 C0
0.034569 14 F6 F4 6B F5 3A 20 89 84 20 8A 5C 08 00 45 C2 00 30 00 01 40 00 40 06 00 00 C0
0.053388 14 F6 F4 6B F5 3A 20 89 84 20 8A 5C 08 00 45 68 00 30 00 01 40 00 40 06 00 00 C0
0.065080 14 F6 F4 6B F5 3A 20 89 84 20 8A 5C 08 00 45 A2 00 30 00 01 40 00 40 06 00 00 C0
0.078190 14 F6 F4 6B F5 3A 20 89 84 20 8A 5C 08 00 45 08 00 30 00 01 40 00 40 06 00 00 C0
0.088730 14 F6 F4 6B F5 3A 20 89 84 20 8A 5C 08 00 45 3C 00 30 00 01 40 00 40 06 00 00 C0
0.102514 14 F6 F4 6B F5 3A 20 89 84 20 8A 5C 08 00 45 C8 00 30 00 01 40 00 40 06 00 00 C0
0.119895 14 F6 F4 6B F5 3A 20 89 84 20 8A 5C 08 00 45 70 00 30 00 01 40 00 40 06 00 00 C0
```

图10 模拟的数据包

4.2.2 数据包解析及特征提取

算法对数据包进行入侵检测时，首先要根据图9的格式解析数据包内容，并把结果保存起来，如版本、头长度、服务类型、总长度、标志位、源IP地址和目标IP地址等。但并不是所有的数据都对入侵检测有用，不同的攻击有不同的特征，我们需要针对各种攻击形式提取出有利于检测的特征来作为算法的输入，从而提高算法的检测效率。

首先，TCP SYN FLOOD攻击是通过大量半连接的SYN报文来耗尽目标计算机的资源，是一种DOS攻击。虽然单个半连接的SYN报文在理论上是正常的，但在单位时间内出现很多半连接的SYN报文就形成了TCP SYN FLOOD攻击。因此，本文提出在每秒时间内记录半连接的SYN包的数量，并将其记作第1个特征 $n_1$。由于分片IP报文攻击会发送大量第一片分片IP报文而没有接下去的分片报文，因此在发生分片IP报文攻击时，单位时间内第一个分片报文的数量和最后一个分片报文的数量存在很大差值，且与正常情况有明显区别。因此，本文采取了每秒记录两者的差值的方法，并把差值记作第2个特征 $n_2$。泪滴攻击的特点是分片报文的偏移字段不正确，即该分片报文的偏移字段的值与上一个分片报文携带的数据长度不匹配，因此这里提出了第3个特征 $p_1$。如果同一个地址发送的分片报文(不是第一个分片报文)的偏移字段的值小于前一个分片报文携带的数据长度，表现出现异常，并记录 $p_1=1$；否则正常，记录 $p_1=0$。Land攻击的特点是源IP地址与目标IP地址相同，因此提出了第4个特征 $p_2$。匹配数据包的源IP地址与目的IP地址，如果相同，表现出现异常，记录 $p_2=1$；否则正常，记录 $p_2=0$。针对WinNuke攻击的特征，如果某个数据包的目标端口是139，138，137，113，535中的一个，且URG位为1，表现出现异常，记录 $p_3=1$；否则正常，记录 $p_3=0$，这是第5个特征。最后是针对TCP标志位的攻击，通常该攻击发生时，数据包TCP标志位是错误设置的，因此需要记录TCP标志位信息作为第6个特征 $P_4$。以上6个特征是对数据包进行解析后提取出来的，分别对应6种攻击特征。

4.2.3 利用NSA检测攻击

有些网络攻击根据单个数据包就可以检测出来(比如针对TCP标志位的攻击)，但是有些网络攻击不能从单个数据包中看出(比如DOS攻击和TCP SYN FLOOD攻击)，必须依靠上下文或根据一段时间内统计的特征才能被检测出来。因此，本文提出了基于NSA的两层检测器来对数据进行入侵检测处理。

第一层是基于数据包的检测。针对在单个数据包上有明显特征的网络攻击，使用前面介绍的4维特征 $p_1$，$p_2$，$p_3$，$p_4$ 对每个数据包进行检测。第二层是基于数据流的检测，针对在单个数据包上没有明显特征的网络攻击，使用每秒从所有数据包中提取出来的特征(即上文提到的 $n_1$，$n_2$)进行检测，每秒检测一次。

下面简单介绍如何使用NSA来检测攻击。根据前文介绍NSA算法只要知道了自体样本集，就可以生成异常检测器来覆盖异常数据，因此可以检测出未知的攻击。在第一层中，数据特征分别为 $p_1(0,1)$，$p_2(0,1)$，$p_3(0,1)$，$p_4$(小于64的整数)，根据前一节的介绍可知自体样本数据为 $p_1=0$，$p_2=0$，$p_3=0$，$p_4\in\{48\ 24\ 20\ 18\ 17\ 16\ 4\ 2\}$；然后，根据自我样本生成第一层检测器。由于自我样本数量少，因此并没有生成自体检测器，而是生成了异常检测器检测器，具体算法流程如图8所示。当生成了基于数据包的异常检测器后，对每一个数据包进行检测，每一个数据包解析后提取出来的 $p_1$，$p_2$，$p_3$，$p_4$ 这4维数据如果与异常检测器匹配，则表示是攻击数据包，否则是正常数据包。

在第二层NSA里，数据特征为 $n_1$(大于0)，$n_2$(大于0)。大部分关于TCP SYN FLOOD攻击检测的文献中都没有直接说明每秒多少个半连接SYN数据包可以定义为TCP SYN FLOOD攻击，有的提出每秒500个SYN包就可以淹没服务器，也有的提到每秒33～100个SYN包。为了区别网络拥堵和网络攻击，特别定义每秒20个以上SYN包就可被标记为TCP SYN FLOOD攻击。因此，第二层NSA算法的自体样本是 $n_1$ 和 $n_2$ 为[0，19]之间的整数，这样有了自我样本后，就可以根据图8的算法生成基于数据流的异常检测器。检测阶段是

把该主机每秒接收和发送的数据包的统计特征 $n_1$ 和 $n_2$ 与第二层的检测器进行匹配，如果被检测器覆盖（即匹配成功），则表示这 1 秒发生了攻击，否则，数据流是正常的。这样，两层的检测器相互合作，就构成了整个基于主机的入侵检测系统。

### 4.3　实验结果

模拟的网络攻击数据包分布如表 1 所列，包括攻击种类、攻击发生的时间和攻击数据包数。在这个基础上，本文利用 NSA 算法生成了两层检测器来检测相应的 6 种攻击。

表 1　模拟的攻击数据包

| 攻击种类 | 高强度 TCP SYN FLOOD | 低强度 TCP SYN FLOOD | 分片 IP 报文 | 泪滴 | Land | WinNuke | 针对 TCP 标志位 |
|---|---|---|---|---|---|---|---|
| 发生时间/s | 40－41 | 10.85－23.35 | 60.85－65.83 | 5.33－30.72 | 66.85－84.85 | 6.33－29.16 | 65.45－68.15 |
| 数据包数/个 | 500 | 500 | 200 | 2 | 10 | 30 | 10 |

第一层基于数据包的检测结果如表 2 所列。可以看出，4 种攻击的数据包总个数为 52，实际检测出的攻击数据包数为 50，检测率为 96.15%，误测率为 0，漏测率为 3.85%。第二层基于数据流的检测结果如表 3 所列。可以看出，2 种攻击的数据包个数为 1200，实际检测出的攻击数据个数为 1154，则检测率为 96.17%，误测率为 0%，漏测率为 3.83%。攻击发生的时间为 10.85 s，40 s 和 60.85 s，而检测到攻击的时间为 11s，40 s 和 61 s，检测时间极短。两层检测器的检测率都高于 96%，误测率为 0，漏测率也低于 3.85%。由此可知，利用 NSA 生成检测器来检测多种攻击时具有较高的性能。

表 2　基于数据包的检测（第一层）

| 异常数据包/个 | 检测出的数据包/个 | 检测率/% | 误测率/% | 漏测率/% |
|---|---|---|---|---|
| 52 | 50 | 96.15 | 0 | 3.85 |

表 3　基于数据流的检测（第二层）

| 实际攻击时间/s | 检测的攻击时间/s | 异常数据包/个 | 检测出的数据包/个 | 检测率/% | 误测率/% | 漏测率/% |
|---|---|---|---|---|---|---|
| 10.85－23.35<br>40－41<br>60.85－65.83 | 11－23<br>40－41<br>61－66 | 1200 | 1154 | 96.17 | 0 | 3.83 |

利用 NSA 生成两层检测器的时间如图 11 所示，由于自体样本数据较少，生成第一层检测器仅需 0.243 s，生成第二层检测器仅需 3.365 s。虽然检测器可以提前生成，并不影响检测攻击的时间，但检测器的生成时间较短也表示算法的性能较好，其运行时间较短。利用两层检测器来检测各种网络攻击数据包的时间如图 12 所示。

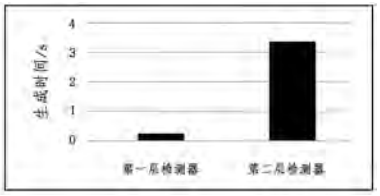
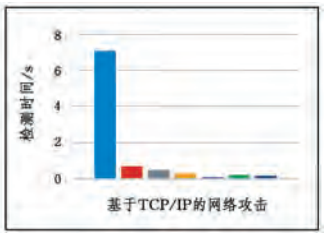



图 11　检测器生成时间的对比




注：从左到右分别为：高强度 SYS FLOOD 攻击，低强度 SYS FLOOD 攻击，分片 IP 报文攻击，泪滴攻击，Land 攻击，WinNuek 攻击，针对 TCP 标志位的攻击

图 12　各种攻击检测时间的对比

其中检测高强度 SYN FLOOD 攻击时，1 秒时间内有 500 个攻击数据包需要要进行解析并检测，数据量比较大，因此检测时间就较长，为 7.086 s；而低强度 SYN FLOOD 攻击和分片 IP 报文攻击都是 1 秒 40 个攻击数据包，检查时间分别为 0.694 s 和 0.48 s，相较于高强度攻击，检测时间就比较短了。而泪滴攻击、Land 攻击、WinNuke 攻击和针对 TCP 标志位的攻击，检测的数据数量为1～3 个，因此检测时间更短，分别为 0.27 s，0.076 s，0.19 s，0.159 s。综上所述，基于 NSA 生成的两层检测器能实时检测网络攻击，检测时间较短。

### 4.4　问题分析

通过模拟的网络攻击入侵检测实验，发现了以下几个问题。

1）模拟的网络攻击种类太少，仅 6 种，导致用到 NSA 算法时数据维度仅为 4 维和 2 维，算法的必要性还没完全体现出来。后期可以增加其他方式的攻击，比如 ICMP 洪水和 UDP 洪水等，增加的攻击方式越多，基于 NSA 算法的入侵检测效果就越明显。

2）已知网络攻击的模式特征还未完全定义，比如对于 TCP SYN FLOOD 攻击，每秒多少半连接 SYN 包可以定义为攻击，或者正常情况下的半连接 SYN 数据包应该在多少范围内。几乎所有的关于 TCP SYN FLOOD 攻击的文献都没有直接定义攻击的模式和特征。

3）实验的数据都由自己模拟，可能不能反映真实网络环境，如果能真实地进行网络攻击并抓包分析，将有利于定义并提取相应攻击的特征。

### 4.5　基于 KDD99 的入侵检测

KDD CUP 99 数据集是从一个模拟的美国空军局域网上采集的 9 个星期的网络连接数据，用于仿真各种用户类型、不同的网络流量和攻击手段。KDD99 是经典的入侵检测数据集。

KDD99 中的数据是以网络连接的形式进行保存的，每个数据都包括 41 个特征属性：3 个符号特征和 38 个数值特征。由于符号特征无法直接进行距离计算，因此需要对符号特征进行十进制编码，将其映射到数值空间。由于 KDD99 数据集里每一维的特征大小都不同，有的差距特别大，如果直接进行距离计算，大数值的特征很可能会掩盖小数值的特征，因此需要进行归一化操作。

KDD99 数据集包括 97278 条 normal 数据和 396743 条 abnormal 数据，数据量较大，因此只选择其中一部分来进行

训练。在实验训练阶段,从 97278 条 normal 数据中分别随机选择 1000,3000 和 5000 条数据来作为训练数据,并用本文提出的基于自体集密度自动划分聚类方法的否定选择算法(DAPC-NSA)来生成异常检测器。在检测阶段,从所有 normal 和 abnormal 的数据中随机提取 6000 条数据作为测试集来进行入侵检测实验,并采用检测器生成时间、入侵检测时间、检测率、误测率作为评价标准来评价算法的性能。实验结果如表 4 所列,本算法在 KDD99 数据集上具有较高的性能,包括较短的检测时间、较高的检测率(即使在只有 1000 条训练数据,检测率也高达 99%)以及较低的误测率。

表 4 基于 KDD99 的入侵检测结果

| 训练数据/条 | 检测器生成时间/s | 检测时间/s | 检测率/% | 误测率/% |
|---|---|---|---|---|
| 1000 | 12.25 | 9.09 | 99.11 | 4.30 |
| 3000 | 36.08 | 25.20 | 99.28 | 3.40 |
| 5000 | 61.41 | 40.37 | 99.16 | 2.38 |

**结束语** 随着互联网的普及,网络安全越来越受到人们的关注。网络中出现了各种各样的攻击,比如 DDOS、蠕虫、病毒等,它们都对网络造成了一定的危害。在网络安全领域,很多研究者也相应地提出了各种方法来检测网络攻击,但是其中大部分方法都只能检测出一类攻击,且检测延迟一般都比较大,不能实时地进行入侵检测。因此,本文提出了一种基于主机的入侵检测方案——基于自体集密度自动划分聚类方法的否定选择算法(DAPC-NSA),来进行网络攻击的入侵检测。本文通过研究网络攻击特征,模拟了 6 种网络攻击数据包,并提出了基于 NSA 的两层检测器来检测攻击。实验表明,该方法不仅可以同时检测出多种网络攻击,而且可以实时检测出网络攻击(检测时间短),同时保证了高检测率和低误检率。虽然由于攻击种类少、数据维度低,本文提出的基于自体集密度自动划分聚类方法的否定选择算法(DAPC-NSA)并没有全部用于模拟的入侵检测实验中,但实验的总体结果较好。如何把 DAPC-NSA 方法真正应用到实际网络入侵检测中,还需要进一步研究。

## 参 考 文 献


[1] XIONG W,HU H N,XIONG N,et al. Anomaly secure detection methods by analyzing dynamic characteristics of the network traffic in cloud communications[J]. Information Scinces,2014(258):403-415.

[2] SPEROTTO A,SCHAFFRATH G,SADRE R,et al. An Overview of IP Flow-Based Intrusion Detection[C]// IEEE Communications Surveys & Tutorials. 2010:343-356.

[3] KIM M S,KONG H J,HONG S C,et al. A Flow-based Method for Abnormal Network Traffic Detection[C]// Proc. IEEE/IFIP Network Network Operations and Management Symposium. 2004:599-612.

[4] TAN Z Y,JAMDAGNI A,HE X,et al. A System for Denial-of-Service Attack Detection Based on Multivariate Correlation Analysis[J]. IEEE Transactions on Parallel and Distributed Systems,2014,25(2):447-456.

[5] IGLESIAS F,ZSEBY T. Analysis of network traffic features for anomaly detection[J]. Machine Learning,2015,101(1-3):59-84.

[6] JYOTHI V,WANG X Y,ADDEPALLI S K,et al. BRAIN:Behavior based Adaptive Intrusion detection in Networks:Using Hardware Performance Counters to detect DDoS Attacks[C]// 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems. 2016:587-588.

[7] CHEN Y,HWANG K,KU W S,et al. Collaborative Detection of DDoS Attacks over Multiple Network Domains[J]. IEEE Transactions on Parallel and Distributed Systems,2007,18(12):1649-1662.

[8] LEE K,KIM J,KWON K H,et al. DDoS attack detection method using cluster analysis[J]. Expert Systems with Applications,2008,34(3):1659-1665.

[9] SIRIS V A,PAPAGALOU F. Application of Anomaly Detection Algorithms for Detecting SYN Flooding Attacks[J]. Computer Communications,2006,29(9):1433-1442.

[10] CHEN W,YEUNG D Y. Defending Against TCP SYN Flooding Attacks Under Different Types of IP Spoofing[C]// Proceedings of the International Conference on Networking. 2006:38.

[11] WANG H N,ZHANG D L,SHIN K G. Detecting SYN Flooding Attacks[C]// IEEE INFOCOM. 2002:1530-1539.

[12] VIS I F A,DE KOSTER R. Transshipment of containers at a container terminal:an overview[J]. European Journal of Operational Research,2003,147(1):1-16.

[13] FORREST S,PERELSON A S,ALLEN L,et al. Self-nonself discrimination in a computer[C]// Proceeding of the IEEE Symposium on Research in Security and Privacy. Oakland:IEEE,1994:202-212.

[14] JI Z. A boundary-aware negative selection algorithm[C]// Proceedings of IASTED International Conference of Artificial Intelligence and Soft Computing(ASC 2005). Spain,2005:379-384.

[15] JI Z,DASGUPTA D. Real-valued negative selection algorithm with variable-sized detectors[M]// Genetic and Evolutionary Computation—GECOO 2004. Springer Berlin Heidelberg,2004:287-298.

[16] ZHOU J,DIPANKAR D. V-detector:An efficient negative selection algorithm with "probablyadequate" detector coverage[J]. Information Sciences,2009,179(10):1390-1406.

[17] GONG M G,ZHANG J,MA J J,et al. An efficient negative selection algorithm with further training for anomaly detection[J]. Knowledge-Based Systems,2012,30(2):185-191.

[18] XU X P,ZHAO P Z. Research on fault data classification based onimproved V- detector algorithm[J]. Application Research of Computers,2013,30(10):2951-2953.

[19] HOQUE N,BHUYAN M H,BAISHYA R C,et al. Network attacks:Taxonomy,tools and systems[J]. Journal of Network and Computer Applications,2014,40(1):307-324.

[20] PILLI E S,JOSHI R C,NIYOGI R. Data Reduction by Identification and Correlation of TCP/IP Attack Attributes for Network Forensics[C]// International Conference and Workshop on Emerging Trends in Technology. 2011:276-283.